

# Managing a Product Development Team

Ken Lehtonen, NASA/GSFC

Larry Barrett, Orbital Sciences Corporation

---

Orbiting 380 miles above the earth, NASA's Hubble Space Telescope (HST) has returned a wealth of scientific data about our universe and galaxies beyond highlighted by spectacular images of the birth and death of stars, colliding galaxies, and other extra-worldly events (See Figure 1.)



Figure 1: Sombrero Shaped Galaxy  
NASA Press Release (10/02/03)

Despite its tremendous success for almost two decades, the HST ground support system experienced down-to-earth problems prior to the turn of the century, namely budgetary ones. To keep HST operating efficiently to 2012 and beyond, the *Vision 2000* project was conceived with the primary goal of substantially reducing the costs of operating and maintaining the spacecraft ground systems. Taking advantage of this atypical management opportunity, a set of Product Development Teams (PDTs) was established, whose charter was to re-engineer the ground system, and in doing so, reduce the remaining life-of-mission operating and maintenance costs, while providing improved reliability and increased capabilities.

This article discusses one of those PDTs, namely the Control Center System (CCS) PDT, which was charged with developing and deploying the system that is still responsible for the overall health and safety of the HST vehicle. Specifically, the Hubble Control Center is tasked with sending commands to the HST vehicle for scientific data acquisition, acquiring real-time engineering telemetry data, and providing accurate and timely problem diagnosis. This

article discusses the overall management of this PDT as it struggled to embrace a brave new world of leading edge technologies and to successfully advance a new management culture. This article focuses on several of the more successful management techniques and strategies that were implemented and that ultimately ensured the success of this team, in short, how we managed project assets —specifically, our people assets.

The major technical goals established for the PDT were:

- To challenge the old ways of doing business and to apply new technologies where appropriate;
- To build a system that combined reused legacy applications (e.g., HST-specific algorithms), Commercial Off the Shelf (COTS) products, Government Off the Shelf (GOTS) products, and to leverage evolving technologies, all within a distributed but scalable architecture;
- To design an evolutionary and maintainable system;
- To execute a development strategy for incremental releases to ensure that the HST operations staff and systems engineers could gain early operations experience, thus giving the development staff time to clarify requirements early in the process; and,
- To become an innovative leader in developing control center systems for NASA-Goddard.

The above five goals became the major guideposts for evaluating how the PDT was grown, how it was managed, and how the technical decision making process, which is always required during the course of a system development project, was optimized.

## LOOKING FOR CHALLENGES

The HST Project team located at Goddard Space Flight Center in Greenbelt, Maryland was responsible for funding the *Vision 2000*

initiative. They made it clear from the outset that they were looking for “new and better ways of doing business,” even if this meant re-engineering the existing ground system and totally replacing it with a new system that was based on advanced and potentially unproven technologies. In effect, they were challenging the PDT to originate a better way of operating and managing the spacecraft, and to replace those pieces of its ground system that did not enable that new concept.

A second, not insignificant challenge was the aggressive schedule that was dictated for this project. It was essential that the new Hubble control center system be operational at least one year before the pending HST Servicing Mission that was scheduled for December 1999. That is, beginning with an intense proof of concept demonstration and culminating with the CCS architecture specification in April 1996, it would be necessary to deliver a fully operational system in less than 2 years. (Note: The existing ground system had taken over 5 years to develop.) This objective drove the delivery of six major software releases within the first year! Figure 2 depicts the cumulative software deliveries in terms of lines of code for that time period.

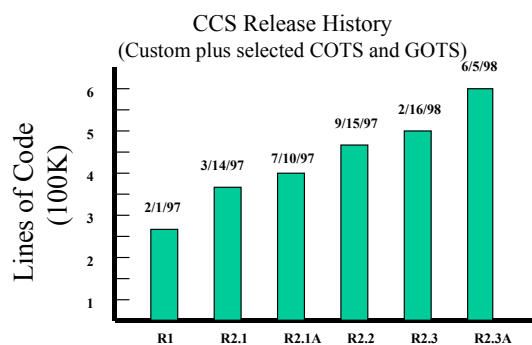


Figure 2. Summary of CCS Productivity

An additional challenge to the management of the PDT was to use initially the legacy software maintenance team who, although highly-trained and well-versed in structured development methodologies (i.e., Yourdon, FORTRAN), were not as skilled in more current system design and development technologies (e.g., OMT[Object Management Technology], C++, Java). In fact, the PDT was “front-loaded” (numbers-wise) with a technical staff that normally wouldn’t be required until after a traditional preliminary design review. The new management team quickly determined that traditional approaches

wouldn’t work as indicated by an attrition rate that approached 30 percent. One management guideline that was actively employed with good results was “utilize those management techniques that had been successfully applied to small teams or were currently being used successfully in similar re-engineering projects.” The goal was to eliminate the sources of inefficiency on the project by building a culture that fostered an atmosphere of cooperation and that was success oriented.

## FLATTENING THE ORGANIZATION

One of the first actions the management team undertook, that gradually paid big dividends, was to flatten the project organization. A minimal project management support staff consisting of two release managers, two quality assurance personnel, a resource scheduler, and a single administrative assistant was established. The remainder of the organizational structure consisted of a set of core technical teams, each with a technical lead “supervisor.” Initially, there was a significant amount of resistance to this “radical” approach, because the traditional hierarchical management structure (“command and control”) from the legacy organization was firmly entrenched. However, the key stakeholder for this project was very supportive of this approach and the staff quickly accepted a structure that imposed less bureaucracy.

Each of the empowered team leads was held responsible for implementing a specific subsystem within the Hubble control center. They were also tasked with ensuring that their staffs were the right size, embodied the appropriate skill mix, and were properly trained. For example, there was a core team to develop the spacecraft command functions, another team to develop the graphical user interface (or GUI), a middleware team, a data management team, etc. The technical decision-making process was “pushed down” as far as possible in order to streamline the overall development effort (remember we had very aggressive schedules to meet). To complete the picture, the release managers, who were ultimately responsible for delivering the next scheduled version of the Hubble control center system, were charged with identifying the resources they needed (hardware, software and staffing) to meet their delivery schedule. Thus, they were required to negotiate with each of the core team leads to borrow

personnel to establish a release team with the right skill mix. Should a conflict arise during this process, then and only then, did the Project Lead intervene to clarify the priorities and/or to reallocate the resources.

As a result of this new organizational structure, team members had both an organizational ‘core’ identity that closely matched their own technical skills, as well as an affiliation with the delivered system release. A better designation for this new organization was “dynamic matrix,” as the free flow of information between and among teams and team members was encouraged, as was staff movement between teams when conditions warranted such a change.

Another facet of this organization’s character was management’s tolerance for “failure” without retribution. After any significant problem was resolved, a post-mortem was conducted, process improvement (if required) was initiated, and/or the team’s skill mix was adjusted (if necessary). This approach led to a project environment that fostered the growth of personal strengths instead of punishing weaknesses. As a result, the staff became more willing to assume responsibility for making the decisions necessary to meet the aggressive schedule. Over time, a set of informal “checks and balances” evolved between the teams that enabled continual progress, rapid decision making, and a reduction in the magnitude of corrections that was required.

#### COMMUNICATION, COMMUNICATION, COMMUNICATION

The CCS management team realized that a project of this size (it started out with 75 people and reached its maximum at about 150 people) and complexity required constant and effective communications—oral as well as written. We took advantage of the fact that the HST Project decided to collocate the majority of the CCS team to an off-site building about ten minutes from the main Goddard campus. Under one roof, we housed the systems engineers, software developers, system testers, hardware engineers, operations personnel (small subset of HST flight operations team), quality assurance, procurement, and management personnel. This arrangement turned out to be one of the major reasons for the success of this project (see “Developing A Cohesive, Cooperative Culture”).

As with any team effort, a combination of both formal and informal communications was required. Formal communications was chiefly used to inform HST stakeholders and senior management of the status of the project. This meant that the core teams and the release managers would provide status estimates for their areas on a weekly basis. The CCS management team would then meet on a scheduled basis with the key project stakeholders and present a consolidated status in terms of schedules, percent complete estimates, and other traditional project management reporting vehicles. Meetings were normally held at Goddard to maintain the buffer between the stakeholders and the development staff. Periodic formal presentations to an independent “audit” team were also required to ensure that all of the PDTs were progressing as planned and in unison with each other.

On an informal basis, the CCS management team implemented a series of actions that proved to be highly beneficial to overall project success. First of all, the building’s layout was leveraged to group each of the technical teams as physically close together as possible. This step enabled significant intra- and inter-team communications at the technical level, as well as for the project team as a whole. For example, all developers of a particular subsystem were either in the same office or were housed in adjacent offices. As a result, it was very easy for anyone on the project to obtain a “real-time” status of the development process, and this staff “mingling” was encouraged and supported by the management team. We also conducted daily 10-minute “stand-up” technical meetings led by the release manager to foster timely communications across groups. To further enhance communications during software release integration, we dedicated a portion of the building as an integration facility. Each Hubble control center core team (e.g., front-end processing, command, GUI) had its own dedicated floor space and workstations. Thus, as the CCS data flows traversed through the system from “left to right” during a particular test, the teams were able to communicate directly and immediately with each other and to identify first hand any interface problems that arose. This was a significant contributing factor to our on-time software deliveries. See Figure 3 below.



**Figure 3: CCS Integration Environment ("The Triangle").**

As the project progressed, it was necessary to increase the size of the current staff, especially in those technology areas where the current legacy personnel were less experienced. About that time, biweekly summer barbecues were also initiated. Each core team, in turn, assumed responsibility for the “theme” of the cookout and for setup and cleanup afterwards. The motivation behind this ‘management-encouraged’ social interaction was the belief that people who got to know each other in an informal, less stressful setting, would work together much more effectively during the stressful software system integration period. This assertion proved to be correct when it was noticed that the traditional “finger pointing” associated with system interface testing was virtually non-existent within the PDT.

It is also important to note that as Project Lead, the government author felt it was necessary to get to know everyone and to conduct frequent, informal MBWA (“management by walking around”) sessions. These sessions enabled him to meet first hand all of the members of the various teams as well as to communicate the ideals for an open, inclusive project, and to encourage the sharing of technical knowledge among team members. It also strengthened the goal of having an active and trusted management presence on the project.

#### AN ELECTRONIC WORLD

Another avenue of communication was the establishment of capabilities to share internal design information efficiently. A conscious decision was made to reduce but not entirely eliminate the need for hardcopy documentation, addressing the often heard comment that as soon

as a document was published it is out of date. To that end, a couple of internal web sites were established that enabled the distribution of PDT documents electronically. These included key management documents and presentations, Master Schedules, OMT and C++ primers, lecture notes, technical tutorials, and individual technical team documentation. Supplementing the web sites was the heavy use and reliance on e-mail. Everyone within the team was outfitted with a personal computer running the same mail package. At the core of the design process, a CASE (Computer Aided Software Engineering) tool was established that was capable of storing our OMT design information electronically and which could be used to generate hardcopy Object-Oriented (O-O) design documents for walkthroughs. This electronic repository was very effective in streamlining the documentation of the development process, but it remained a key engineering component during the transition from the purely development project to the current sustaining engineering environment.

In effect what was accomplished was to embed an effective information management environment within the project. By carefully selecting and tailoring the right tools and processes, technical communication was significantly facilitated and the information needs of the project were met. Because of this tailoring, we had “at our fingertips” the necessary information to facilitate decision making processes—as a direct result, quick analysis of alternatives and timely selection were possible, which kept the team moving forward at all times. Because of the availability of all design information, post-mortem processes became more efficient, timely, and accurate.

In summary, the intelligent use of electronic communications and engineering tools became another factor contributing to streamlining the management process and enhancing peer-to-peer interaction.

#### GROWING THE TEAM

As previously mentioned, one of the original project constraints was to use a software staff that was highly skilled in developing and maintaining FORTRAN code on Digital VMS systems. However, despite our doubts about the technical currency of this legacy team, they embodied the Hubble domain knowledge that

was both critical and necessary to development of the new control center. Therefore, a strategy needed to be put in place to enhance the effectiveness of this legacy staff.

Since a decision had been made to develop the control center using O-O technology targeting a Unix environment, the challenge became one of “converting” as many legacy programmers as possible to the OMT methodology. One of the greatest obstacles to achieving this goal was the legacy staff itself. Specifically, this team had to be convinced that there was both a personal and programmatic benefit to transitioning their design skills from Structured Analysis to OMT and their implementation target from FORTRAN to C++/Java. The management team was able to demonstrate these benefits through a series of technical briefings that demonstrated the additional capabilities and flexibility of the O-O technologies. These briefings served to convince senior members of the legacy staff that the HST project would benefit from employing modern software design principles, such as object oriented programming, in order to develop a system that had to last at least another decade.

At this point, the green light was given by HST senior management to go on a hiring “binge” to acquire additional staff with key O-O and C++/Java skills (being a team player was also an important hiring consideration). A major objective was to use these new members to bring the legacy staff up to the necessary level of technical capability. This was accomplished through the following multi-faceted training approach:

- 1) A massive just-in-time training effort for the whole development team was initiated (at this point the architecture of the Hubble control center system was just about completed). This was done by bringing in-house some of the top O-O trainers in the nation to provide targeted training. The traditional training approach was reversed by first training the team in the C++ language specifics (they were already familiar with FORTRAN and in some cases the ‘C’ language) and then providing on-site training courses in generalized Object-Oriented Analysis and Object-Oriented Design. This approach worked better because the staff was more comfortable with implementation technologies from which

they could then abstract the methodological underpinnings;

- 2) For those major COTS products that were selected into the architecture of the new system, the vendors were willing to train the team in the specifics of their products;
- 3) To supplement the standard classroom training, technical consultants (see below) were brought in who not only mentored the team but were exemplar software developers in their own right;
- 4) To improve our contacts with outside industry, the staff was encouraged to attend technical conferences and to present papers or provide demonstrations of the CCS technologies under development; and,
- 5) Internal technical demonstrations were scheduled of mature software, not only for the CCS staff but for Goddard senior management as well. This was not only a morale boost for the presenters, but provided another means for communicating technical information throughout the development team and to the stakeholders back at Goddard. A side benefit of the demonstrations was that they helped identify specific technical skills of project teammates to the rest of the staff.

In spite of the focused training effort, it became apparent during our design activities that we needed to sprinkle our emerging O-O team with some experienced on-site O-O and C++ expertise. With senior management’s approval, and as part of the CCS management philosophy to engage outside expertise, we contracted with an expert organization with a proven track record in O-O development to provide a small number of on-site consultants. To avoid the traditional (often contentious) consultant-client relationship, the management team decided to rapidly assimilate these consultants into our own evolving culture and make them an active part of the CCS team. For their part, the consultants provided mentoring services on a one-to-one basis in analysis & design, C++ language skills, software debugging, and, in general, helped the team gain the necessary O-O technical skills and confidence. We also made them an integral part of the development team by assigning them key pieces of application software to design and code. (One of the consultants was tasked to lead the Middleware Team until a permanent replacement could be found.) The truth of the matter was that these consultants were

instrumental to the successful, on-time delivery of the Release 1 system, and provided a significant return on investment for their services.

#### DEVELOPING A COHESIVE, COOPERATIVE CULTURE

By being physically separate from the Goddard mainstream, the PDT was able to develop its own unique management culture and style that fit the environment and the goals that it set out to achieve. One of the first management goals was to replace the typical atmosphere of competition between contractors and animosity with the customer with a more universally cooperative environment. This transformation was attempted by restructuring the team dynamics by implementing and expanding upon the concept of a “badgeless team.”

The badgeless team concept meant breaking down traditional barriers and roles (often contractual) between civil servants (HST being a Government-run project) and contractor personnel, as well as among a variety of support contractors (there were eventually over a dozen different companies represented on this PDT). A bigger challenge however turned out to be convincing the various contractor and government supervisors that such an approach would work. In actual practice, there were civil servants reporting to contractor personnel; contractors reporting to contractors of the same company; and, contractors reporting to contractors of different companies. This represented a radical departure from what senior NASA management viewed as the way civil servants and contractors were supposed to relate to each other. It should be noted that this was an evolutionary process since not everyone on the CCS PDT was comfortable with this new management philosophy, and some chose to leave the project.

In retrospect, this management strategy became one of the main reasons the team was so successful. The emphasis on technical achievements and having a shared vision along with a “laser-like” focus on the CCS goals instead of which particular company should get the credit for the work accomplished enabled a unique situation. The ultimate goal was to erase from people’s mindset the process of “going through channels.” Everything you needed to

get your job done was resident at the collocation facility (lovingly referred to by the staff as just Colo). Again, the relative physical isolation enabled the staff to significantly reduce (but not entirely eliminate) traditional corporate politics and jurisdictional disputes that previously had hindered close, technical exchanges and cooperation between different companies working on the same project.

To achieve this cohesive, cooperative culture, the management team recognized early on that the internal “naysayers” needed to either be converted or to be strongly encouraged to leave the project. Teamwork and the free exchange of ideas were to be the hallmarks of this project. Over time these radical ideas bore fruit, as team members felt comfortable in creating ad hoc teams in the hallways and freely circulated around the building. The staff developed a high measure of trust between management and team personnel as well as between and among the individual teams. There was an overall collegial, community atmosphere that pervaded this project that allowed all members of the CCS project to excel and exceed all expectations: from both a technical and personal perspective.

#### MANAGEMENT PRINCIPLES: IMPLEMENTATION STRATEGIES

Besides the aforementioned management principles that provided the greatest benefit in managing the CCS PDT technical staff, the following management principles (listed in no particular order) also served to sustain the high productivity environment. Some of these principles are obvious; some are espoused in current management science texts; and, some will only work in a collocated environment.

1. Use Integrated Product Teams to provide short-term results. A meta-goal of every project is to make good design decisions and to develop the corresponding products as quickly as possible. The CCS PDT management chose to select from the entire organization those persons who could best produce a particular product, assembled them into a small team, gave them the authority to make the necessary decisions, and, when the product was completed, returned them back to their core technical teams. (“Now you see them/now you

don't") Examples of these teams within the CCS project included the Top Down Architecture Team, Data Format Team, and Automation Working Group. These were all ad hoc high-performance teams, who successfully created essential design products from which the remainder of the CCS architecture evolved.

2. Use the "80-20" Rule. As is the case with most projects, the CCS PDT existed in a very dynamic environment where technology was rapidly evolving and user requirements were negotiable. Recognizing this, a decision was made to expedite the decision-making process and to avoid "paralysis by analysis," by employing the 80/20 rule. For example, if a COTS product could be found that satisfied at least 80% of the target user requirements, then feedback from the users would be solicited to determine if this was adequate. This process was driven by the understanding that not all user requirements are created equal and thus, implementation of the least important 20% can often be deferred, sometimes indefinitely. This process also served to keep the user community involved in critical design decisions so they remained 'part of the solution.'
3. Establish Proof of Concept and/or Prototyping Teams. Early on, the core technical teams were tasked with performing risk mitigation activities while the final architecture of the control center was being hammered out. (Remember: the team was originally front-loaded with a legacy software staff.) The Proof-of-Concept Team (POC) was instrumental in identifying and demonstrating promising new technologies (e.g., Java applets, collaborative tools, and COTS packages). The results were feed back to the Top-Down Architecture Team to help justify and substantiate the proposed control center architecture. This served as an excellent risk mitigation activity by introducing the staff to a significant number of new (and sometimes unproven) technologies. One of the PDT's primary objectives was to leverage COTS hardware and software solutions as much as feasible, and thus, many of the teams worked to prototype these packages in an environment as close to

that envisioned for the actual control center. Out of these prototyping activities emerged a suite of COTS (and GOTS) solutions that was later integrated into the control center design, with the added benefit of reducing both risk and implementation time.

4. Implement a "rewards and awards" program. After each successful delivery of a Control Center System release, the Project Lead personally acknowledged each individual who contributed to that release with a Kudos® candy bar. This informal award was so well received that the team members came to expect one of these visits right after each software delivery. This was one strategy that cost so little but where the gains were immeasurable. The PDT was also very fortunate in having an upper management team at Goddard that was very supportive of both individual and team efforts. They funded an "incentive awards" program that rewarded the Hubble control center team members with bonus checks upon a successful software delivery (instead of just funding the prime contractor's award fee).
5. Integrate and elevate traditionally background activities into the main software development cycle. The PDT recognized the importance and value of traditional support functions to the successful development and deployment of the control center system. Four examples illustrate this concept: (1) The Infrastructure Team provided the systems administration, networking, and hardware expertise necessary to define the overall system topology and operations concept. (2) The Quality Assurance Team was responsible for ensuring that processes were followed and that design and coding standards were adhered to during all phases of development. (3) The Methodology Team was responsible for tailoring and maintaining the CASE tool used to capture all of the design information for the developers. (4) The Configuration and Change Management Team developed the electronic tools necessary to support our software baseline control process (configuration management) and the rapid capture and dissemination of problem reports (change management).

6. Hire college students for the summer. Here was an overlooked area that paid big dividends for this PDT. The team was fortunate to employ three college interns who were able to contribute significantly to the development process. Specifically, these summer interns contributed to the conversion of the command subsystem from VMS to Unix; developed and tested Java applets for the GUI subsystem; and, developed performance benchmarks for a new tape-based archive system that had been procured. The interns were treated as if they were part of the overall team, were challenged technically, and helped the PDT to maintain an optimal skill mix for the tasks that were at hand.
7. Establish a mechanism for detecting and resolving conflict as quickly as possible. Conflict is inevitable no matter the size of the team or its objectives. Establishing mechanisms to deal with the various forms of conflict is critical to the success of any team. In this case, specific technical issues that cut across core team boundaries, were referred to the Control Center System Architecture Board (CAB), which was chaired by the Lead Systems Engineer. All issues related to the architecture, design, implementation, and correction of the control center software were also referred to the CAB for resolution. Intra-team conflicts were expected to be resolved within the specific core team boundaries. At any time, a member of a core team could refer unresolved non-technical conflicts directly to the Project Management Team. In these cases, the staff member's company supervisor could be included in the process to ensure a timely and equitable resolution.

## FINAL REMARKS

Because of the relative isolation from its predecessor culture, the Hubble Control Center System PDT management team was granted a great degree of latitude in applying unconventional management techniques. The goals of the management team were no different than those of most systems development projects.

These goals being:

1. To establish an organization structure that provides the right level of control without impeding progress;
2. To establish and maintain a high level of morale that fosters a 'team identity';
3. To allocate project resources in a balanced manner;
4. To intelligently manage technical and non-technical (e.g., schedule, cost) risk;
5. To leverage the existing skill set of the staff while continuing to cultivate weaker areas;
6. To acquire accurate and timely status of the overall project as well as each sub-element;
7. To meet or exceed expected productivity estimates;
8. To develop and deliver a high quality product to the customer;
9. To empower the staff to make timely and accurate design decisions to minimize rework;
10. To institute a method of achieving internal process improvement;
11. To enable synergy and a spirit of cooperation within the project; and,
12. To quickly detect and resolve internal conflict.

Many of the management techniques discussed in this paper contributed to the achievement of one or more of these management goals.

Table 1 summarizes this information. Each column represents one of the management goals itemized in the previous list. The rows identify key management actions that have been presented throughout the main body of this paper. Marks in the table indicate those management actions that directly or indirectly contributed to the satisfaction of the corresponding goal. It should be noted that these marks represent the assessments of the authors and were not measured using any formal metrics.

In summary, despite the progress made over the last twenty-five years in advancing the state of system and software engineering practices, including improved methodologies, new languages, visual tools, online debuggers, lightning-fast PCs, and CASE tools, project success still comes down to people. Management still needs to find the best ones available, or be willing to invest the time and training dollars in the current staff. Once an exceptional staff is in place, it is necessary to

keep that team focused on the technical milestones (eliminate the politics if possible) as well as to provide a means of recognition through something as simple as a Kudos bar to a full-scale incentive bonus.

The Hubble control center PDT management team undertook all of these actions and were rewarded with a highly skilled, productive, cohesive, and communicative staff with an

attrition rate that was significantly less than industry norms of the time. However, like all good things acquired, there is an upkeep cost: people need technical challenges, opportunities for additional training and professional growth, and a little TLC and recognition every now and then. But the results are well worth it, and, besides, you can't be successful without them!

Table 1: Management Actions Contributing to Goal Satisfaction

|  | Organizational Structure | Team Identity | Balance Resources | Manager Risk | Leverage Staff Skills | Accurate Status | High Productivity | High Quality Product | Minimize Rework | Process Improvement | Staff Synergy | Conflict Resolution |
|--|--------------------------|---------------|-------------------|--------------|-----------------------|-----------------|-------------------|----------------------|-----------------|---------------------|---------------|---------------------|
| Flatten Organizational Structure             | •                        |               |                   |              |                       | •               | •                 | •                    |                 |                     | •             | •                   |
| Empower Staff                                |                          | •             |                   |              | •                     |                 | •                 | •                    |                 |                     | •             |                     |
| Create 'Dynamic Matrix'                      | •                        |               | •                 |              |                       |                 |                   |                      | •               |                     |               |                     |
| Allow Failure without Retribution            |                          |               |                   |              | •                     |                 |                   |                      |                 | •                   |               | •                   |
| Perform Post-Mortems                         |                          |               |                   | •            |                       |                 | •                 |                      | •               | •                   |               |                     |
| Collocate Staff                              | •                        | •             | •                 |              |                       | •               | •                 | •                    | •               |                     |               |                     |
| Encourage Internal Communication             |                          | •             |                   | •            | •                     | •               | •                 | •                    | •               |                     | •             |                     |
| Sponsor Social Events (Barbecues)            |                          | •             |                   |              |                       |                 | •                 | •                    |                 |                     |               | •                   |
| Management By Walking Around                 |                          |               | •                 | •            | •                     | •               |                   |                      |                 | •                   |               | •                   |
| Up-to-date Design Information on Web         |                          |               |                   | •            |                       | •               | •                 |                      | •               |                     | •             |                     |
| Mandatory CASE Tool Usage                    |                          |               |                   | •            |                       |                 | •                 | •                    | •               |                     |               |                     |
| Multi-faceted Staff Training (Formal, OJT)   |                          | •             |                   | •            | •                     |                 |                   | •                    |                 | •                   |               |                     |
| External Technical Interaction (Conferences) |                          |               |                   |              | •                     |                 |                   |                      |                 | •                   |               |                     |
| Internal Technical Demos                     |                          | •             |                   | •            | •                     |                 |                   | •                    |                 | •                   | •             |                     |
| Establish 'Badgeless' Team                   | •                        | •             |                   |              | •                     |                 |                   |                      |                 |                     | •             |                     |
| Remove 'Naysayers'                           |                          | •             |                   |              |                       |                 | •                 | •                    |                 |                     | •             | •                   |
| Use Integrated Product Teams                 |                          |               | •                 | •            | •                     |                 | •                 | •                    | •               |                     |               |                     |
| 80-20 Rule                                   |                          |               |                   |              |                       |                 |                   |                      | •               |                     |               |                     |
| Actively Perform Prototyping                 |                          |               |                   | •            | •                     |                 |                   | •                    |                 |                     |               |                     |
| Provide Rewards and Awards                   |                          | •             |                   | •            |                       |                 | •                 | •                    |                 | •                   |               | •                   |



**Figure 4: Release of HST After Servicing Mission 2**